



\*\*FILE\*\*iD\*\*SYSCVRTIM

F 5

S1  
VC

(1)	218	CONVERT BINARY TIME TO ASCII STRING
(1)	313	CONVERT ASCII STRING TO BINARY TIME
(1)	580	CONVERT BINARY TIME TO NUMERIC TIME

0000 1 .TITLE SYSCVRTIM - SYSTEM SERVICES TO CONVERT TIME  
0000 2 .IDENT 'V04-000'  
0000 3  
0000 4 :  
0000 5 \*\*\*\*\*  
0000 6 \*  
0000 7 \* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0000 8 \* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0000 9 \* ALL RIGHTS RESERVED.  
0000 10 \*  
0000 11 \* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0000 12 \* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0000 13 \* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0000 14 \* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0000 15 \* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0000 16 \* TRANSFERRED.  
0000 17 \*  
0000 18 \* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0000 19 \* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0000 20 \* CORPORATION.  
0000 21 \*  
0000 22 \* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0000 23 \* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0000 24 \*  
0000 25 \*  
0000 26 \*\*\*\*\*  
0000 27  
0000 28 D. N. CUTLER 6-JAN-76  
0000 29  
0000 30 SYSTEM SERVICES TO CONVERT TIME  
0000 31  
0000 32 CONVERT BINARY TIME TO ASCII STRING  
0000 33 CONVERT ASCII STRING TO BINARY TIME  
0000 34 CONVERT BINARY TIME TO NUMERIC FORMAT  
0000 35  
0000 36 THE CONVERSION ALGORITHMS USED HEREIN WERE DEVELOPED BY P. CONKLIN,  
0000 37 M. SPIER, AND D. ROSENBERY ON THE PDP-10.  
0000 38  
0000 39 MODIFIED BY:  
0000 40  
0000 41 V03-001 KDM0086 Kathleen D. Morse 02-Apr-1982  
0000 42 Correctly acquire system time, even in case where  
0000 43 secondary processor is accessing EXESG0 SYSTIME while  
0000 44 the primary processor is updating it (1T/782 case).  
0000 45  
0000 46 V02-004 ROW37307 Ralph O. Weber 27-Jul-1981  
0000 47 Fix EXESBINTIM to treat decimal point preceding hundredths of  
0000 48 a second field as a true decimal point. IE: to cause 0:0:0.1  
0000 49 to convert to 1 tenth of a second rather than to 1 hundredth  
0000 50 of a second. Also allow indefinite length fractional value  
0000 51 fields. Use the thousandths digit to round the hundredths  
0000 52 value, and ignore all digits following the thousandths digit.  
0000 53 The entire field, upto the first trailing blank, is still  
0000 54 processed. Therefore, non-numeric characters in the  
0000 55 fractional seconds field will still produce an Invalid Time  
0000 56 return code.  
0000 57 :

0000 58 : V02-003 TCM0001 Trudy C. Matthews 03-Jun-1981  
0000 59 : Fix CONVERT subroutine in EXESBINTIM to ignore blanks. This  
0000 60 : fix allows trailing blanks after a truncated time field.  
0000 61 :  
0000 62 :  
0000 63 :  
0000 64 : MACRO LIBRARY CALLS  
0000 65 :  
0000 66 :  
0000 67 : SSSDEF ;DEFINE SYSTEM STATUS VALUES  
0000 68 :  
0000 69 :  
0000 70 : LOCAL SYMBOLS  
0000 71 :  
0000 72 : ARGUMENT LIST OFFSET DEFINITIONS FOR CONVERT BINARY TIME TO ASCII STRING  
0000 73 :  
0000 74 :  
00000004 0000 75 ATIMLEN=4 :ADDRESS OF WORD TO STORE LENGTH  
00000008 0000 76 ATIMBUF=8 :ADDRESS OF OUTPUT BUFFER DESCRIPTOR  
0000000C 0000 77 ATIMADR=12 :ADDRESS OF 64-BIT ABSOLUTE OR DELTA TIME  
00000010 0000 78 ACVTFLG=16 :CONVERSION INDICATOR  
0000 79 :  
0000 80 : ARGUMENT LIST OFFSET DEFINITIONS FOR CONVERT ASCII STRING TO BINARY TIME  
0000 81 :  
0000 82 :  
00000004 0000 84 BTIMBUF=4 :ADDRESS OF ASCII STRING DESCRIPTOR  
00C00008 0000 85 BTIMADR=8 :ADDRESS TO STORE 64-BIT ABSOLUTE OR DELTA T  
0000 86 :  
0000 87 : ARGUMENT LIST OFFSET DEFINITIONS FOR CONVERT BINARY TIME TO NUMERIC TIME  
0000 88 :  
0000 89 :  
0000 90 :  
00000004 0000 91 NTIMBUF=4 :ADDRESS OF 7-WORD BUFFER TO RECEIVE TIME  
00000008 0000 92 NTIMADR=8 :ADDRESS OF 64-BIT ABSOLUTE OR DELTA TIME  
0000 93 :  
0000 94 : CONVERSION CONSTANTS  
0000 95 :  
0000 96 : TOTAL DAYS IN A CENTURY  
0000 97 :  
0000 98 :  
0000 99 :  
00008EAC 0000 100 CENTURYDAYS=<100\*365>+<100/4>-<100/100> ;  
0000 101 :  
0000 102 :  
0000 103 : AVERAGE QUARTER DAYS PER CENTURY  
0000 104 :  
0000 105 :  
00023AB1 0000 106 QDAYSPCENT=<<100\*365>+<100/4>-<100/100>>\*4>+<400/400> ;  
0000 107 :  
0000 108 :  
0000 109 : AVERAGE QUARTER DAYS PER YEAR  
0000 110 :  
0000 111 :  
000005B5 0000 112 QDAYSPYEAR=<365\*4>+1 :  
0000 113 :  
0000 114 :  
0000 115 :  
0000 116 :  
0000 117 :  
0000 118 :  
0000 119 :  
0000 120 :  
0000 121 :  
0000 122 :  
0000 123 :  
0000 124 :  
0000 125 :  
0000 126 :  
0000 127 :  
0000 128 :  
0000 129 :  
0000 130 :  
0000 131 :  
0000 132 :  
0000 133 :  
0000 134 :  
0000 135 :  
0000 136 :  
0000 137 :  
0000 138 :  
0000 139 :  
0000 140 :  
0000 141 :  
0000 142 :  
0000 143 :  
0000 144 :  
0000 145 :  
0000 146 :  
0000 147 :  
0000 148 :  
0000 149 :  
0000 150 :  
0000 151 :  
0000 152 :  
0000 153 :  
0000 154 :  
0000 155 :  
0000 156 :  
0000 157 :  
0000 158 :  
0000 159 :  
0000 160 :  
0000 161 :  
0000 162 :  
0000 163 :  
0000 164 :  
0000 165 :  
0000 166 :  
0000 167 :  
0000 168 :  
0000 169 :  
0000 170 :  
0000 171 :  
0000 172 :  
0000 173 :  
0000 174 :  
0000 175 :  
0000 176 :  
0000 177 :  
0000 178 :  
0000 179 :  
0000 180 :  
0000 181 :  
0000 182 :  
0000 183 :  
0000 184 :  
0000 185 :  
0000 186 :  
0000 187 :  
0000 188 :  
0000 189 :  
0000 190 :  
0000 191 :  
0000 192 :  
0000 193 :  
0000 194 :  
0000 195 :  
0000 196 :  
0000 197 :  
0000 198 :  
0000 199 :  
0000 200 :  
0000 201 :  
0000 202 :  
0000 203 :  
0000 204 :  
0000 205 :  
0000 206 :  
0000 207 :  
0000 208 :  
0000 209 :  
0000 210 :  
0000 211 :  
0000 212 :  
0000 213 :  
0000 214 :  
0000 215 :  
0000 216 :  
0000 217 :  
0000 218 :  
0000 219 :  
0000 220 :  
0000 221 :  
0000 222 :  
0000 223 :  
0000 224 :  
0000 225 :  
0000 226 :  
0000 227 :  
0000 228 :  
0000 229 :  
0000 230 :  
0000 231 :  
0000 232 :  
0000 233 :  
0000 234 :  
0000 235 :  
0000 236 :  
0000 237 :  
0000 238 :  
0000 239 :  
0000 240 :  
0000 241 :  
0000 242 :  
0000 243 :  
0000 244 :  
0000 245 :  
0000 246 :  
0000 247 :  
0000 248 :  
0000 249 :  
0000 250 :  
0000 251 :  
0000 252 :  
0000 253 :  
0000 254 :  
0000 255 :  
0000 256 :  
0000 257 :  
0000 258 :  
0000 259 :  
0000 260 :  
0000 261 :  
0000 262 :  
0000 263 :  
0000 264 :  
0000 265 :  
0000 266 :  
0000 267 :  
0000 268 :  
0000 269 :  
0000 270 :  
0000 271 :  
0000 272 :  
0000 273 :  
0000 274 :  
0000 275 :  
0000 276 :  
0000 277 :  
0000 278 :  
0000 279 :  
0000 280 :  
0000 281 :  
0000 282 :  
0000 283 :  
0000 284 :  
0000 285 :  
0000 286 :  
0000 287 :  
0000 288 :  
0000 289 :  
0000 290 :  
0000 291 :  
0000 292 :  
0000 293 :  
0000 294 :  
0000 295 :  
0000 296 :  
0000 297 :  
0000 298 :  
0000 299 :  
0000 300 :  
0000 301 :  
0000 302 :  
0000 303 :  
0000 304 :  
0000 305 :  
0000 306 :  
0000 307 :  
0000 308 :  
0000 309 :  
0000 310 :  
0000 311 :  
0000 312 :  
0000 313 :  
0000 314 :  
0000 315 :  
0000 316 :  
0000 317 :  
0000 318 :  
0000 319 :  
0000 320 :  
0000 321 :  
0000 322 :  
0000 323 :  
0000 324 :  
0000 325 :  
0000 326 :  
0000 327 :  
0000 328 :  
0000 329 :  
0000 330 :  
0000 331 :  
0000 332 :  
0000 333 :  
0000 334 :  
0000 335 :  
0000 336 :  
0000 337 :  
0000 338 :  
0000 339 :  
0000 340 :  
0000 341 :  
0000 342 :  
0000 343 :  
0000 344 :  
0000 345 :  
0000 346 :  
0000 347 :  
0000 348 :  
0000 349 :  
0000 350 :  
0000 351 :  
0000 352 :  
0000 353 :  
0000 354 :  
0000 355 :  
0000 356 :  
0000 357 :  
0000 358 :  
0000 359 :  
0000 360 :  
0000 361 :  
0000 362 :  
0000 363 :  
0000 364 :  
0000 365 :  
0000 366 :  
0000 367 :  
0000 368 :  
0000 369 :  
0000 370 :  
0000 371 :  
0000 372 :  
0000 373 :  
0000 374 :  
0000 375 :  
0000 376 :  
0000 377 :  
0000 378 :  
0000 379 :  
0000 380 :  
0000 381 :  
0000 382 :  
0000 383 :  
0000 384 :  
0000 385 :  
0000 386 :  
0000 387 :  
0000 388 :  
0000 389 :  
0000 390 :  
0000 391 :  
0000 392 :  
0000 393 :  
0000 394 :  
0000 395 :  
0000 396 :  
0000 397 :  
0000 398 :  
0000 399 :  
0000 400 :  
0000 401 :  
0000 402 :  
0000 403 :  
0000 404 :  
0000 405 :  
0000 406 :  
0000 407 :  
0000 408 :  
0000 409 :  
0000 410 :  
0000 411 :  
0000 412 :  
0000 413 :  
0000 414 :  
0000 415 :  
0000 416 :  
0000 417 :  
0000 418 :  
0000 419 :  
0000 420 :  
0000 421 :  
0000 422 :  
0000 423 :  
0000 424 :  
0000 425 :  
0000 426 :  
0000 427 :  
0000 428 :  
0000 429 :  
0000 430 :  
0000 431 :  
0000 432 :  
0000 433 :  
0000 434 :  
0000 435 :  
0000 436 :  
0000 437 :  
0000 438 :  
0000 439 :  
0000 440 :  
0000 441 :  
0000 442 :  
0000 443 :  
0000 444 :  
0000 445 :  
0000 446 :  
0000 447 :  
0000 448 :  
0000 449 :  
0000 450 :  
0000 451 :  
0000 452 :  
0000 453 :  
0000 454 :  
0000 455 :  
0000 456 :  
0000 457 :  
0000 458 :  
0000 459 :  
0000 460 :  
0000 461 :  
0000 462 :  
0000 463 :  
0000 464 :  
0000 465 :  
0000 466 :  
0000 467 :  
0000 468 :  
0000 469 :  
0000 470 :  
0000 471 :  
0000 472 :  
0000 473 :  
0000 474 :  
0000 475 :  
0000 476 :  
0000 477 :  
0000 478 :  
0000 479 :  
0000 480 :  
0000 481 :  
0000 482 :  
0000 483 :  
0000 484 :  
0000 485 :  
0000 486 :  
0000 487 :  
0000 488 :  
0000 489 :  
0000 490 :  
0000 491 :  
0000 492 :  
0000 493 :  
0000 494 :  
0000 495 :  
0000 496 :  
0000 497 :  
0000 498 :  
0000 499 :  
0000 500 :  
0000 501 :  
0000 502 :  
0000 503 :  
0000 504 :  
0000 505 :  
0000 506 :  
0000 507 :  
0000 508 :  
0000 509 :  
0000 510 :  
0000 511 :  
0000 512 :  
0000 513 :  
0000 514 :  
0000 515 :  
0000 516 :  
0000 517 :  
0000 518 :  
0000 519 :  
0000 520 :  
0000 521 :  
0000 522 :  
0000 523 :  
0000 524 :  
0000 525 :  
0000 526 :  
0000 527 :  
0000 528 :  
0000 529 :  
0000 530 :  
0000 531 :  
0000 532 :  
0000 533 :  
0000 534 :  
0000 535 :  
0000 536 :  
0000 537 :  
0000 538 :  
0000 539 :  
0000 540 :  
0000 541 :  
0000 542 :  
0000 543 :  
0000 544 :  
0000 545 :  
0000 546 :  
0000 547 :  
0000 548 :  
0000 549 :  
0000 550 :  
0000 551 :  
0000 552 :  
0000 553 :  
0000 554 :  
0000 555 :  
0000 556 :  
0000 557 :  
0000 558 :  
0000 559 :  
0000 560 :  
0000 561 :  
0000 562 :  
0000 563 :  
0000 564 :  
0000 565 :  
0000 566 :  
0000 567 :  
0000 568 :  
0000 569 :  
0000 570 :  
0000 571 :  
0000 572 :  
0000 573 :  
0000 574 :  
0000 575 :  
0000 576 :  
0000 577 :  
0000 578 :  
0000 579 :  
0000 580 :  
0000 581 :  
0000 582 :  
0000 583 :  
0000 584 :  
0000 585 :  
0000 586 :  
0000 587 :  
0000 588 :  
0000 589 :  
0000 590 :  
0000 591 :  
0000 592 :  
0000 593 :  
0000 594 :  
0000 595 :  
0000 596 :  
0000 597 :  
0000 598 :  
0000 599 :  
0000 600 :  
0000 601 :  
0000 602 :  
0000 603 :  
0000 604 :  
0000 605 :  
0000 606 :  
0000 607 :  
0000 608 :  
0000 609 :  
0000 610 :  
0000 611 :  
0000 612 :  
0000 613 :  
0000 614 :  
0000 615 :  
0000 616 :  
0000 617 :  
0000 618 :  
0000 619 :  
0000 620 :  
0000 621 :  
0000 622 :  
0000 623 :  
0000 624 :  
0000 625 :  
0000 626 :  
0000 627 :  
0000 628 :  
0000 629 :  
0000 630 :  
0000 631 :  
0000 632 :  
0000 633 :  
0000 634 :  
0000 635 :  
0000 636 :  
0000 637 :  
0000 638 :  
0000 639 :  
0000 640 :  
0000 641 :  
0000 642 :  
0000 643 :  
0000 644 :  
0000 64

00000000 0000 115 : TOTAL DAYS IN A QUADRICENTURY  
 00000000 0000 116 :  
 00023AB1 0000 117 :  
 00000000 0000 118 QUADRIDAYS=<400\*365>+<400/4>-<400/100>+<400/400> ;  
 00000000 0000 119 :  
 00000000 0000 120 :  
 00000000 0000 121 : TOTAL DAYS IN A QUADYEAR  
 00000000 0000 122 :  
 00000000 0000 123 :  
 0000005BS 0000 124 QUADYEARDAYS=<365\*4>+1 :  
 00000000 0000 125 :  
 00000000 0000 126 :  
 00000000 0000 127 : OFFSET IN DAYS FROM 1-JAN-1501 TO 17-NOV-1858  
 00000000 0000 128 :  
 00000000 0000 129 :  
 0001FE98 0000 130 TIMOFF1=<<1858-1501>\*365>+<<1858-1501>/4>-<<1858-1501>/100>+<<1858-1501>/400>+ - ;  
 00000000 0000 131 31+28+31+30+31+30+31+31+31+17 ;  
 00000000 0000 132 :  
 00000000 0000 133 :  
 00000000 0000 134 : OFFSET IN DAYS FROM 1-JAN-1601 TO 17-NOV-1858  
 00000000 0000 135 :  
 00000000 0000 136 :  
 00016FEC 0000 137 TIMOFF2=<<1858-1601>\*365>+<<1858-1601>/4>-<<1858-1601>/100>+<<1858-1601>/400>+ - ;  
 00000000 0000 138 31+28+31+30+31+30+31+31+30+31+17 ;  
 00000000 0000 139 :  
 00000000 0000 140 :  
 00000000 0000 141 : CHARACTER CODE DEFINITIONS  
 00000000 0000 142 :  
 00000000 0000 143 :  
 00000000 0000 144 BLANK=32 :  
 00000000 0000 145 COLON=58 :  
 00000000 0000 146 HYPHEN=45 :  
 00000000 0000 147 NINE=57 :  
 00000000 0000 148 ONE=48 :  
 00000000 0000 149 PERIOD=46 :  
 00000000 0000 150 :  
 00000000 0000 151 :  
 00000000 0000 152 : NUMERIC TIME BUFFER OFFSET DEFINITIONS  
 00000000 0000 153 :  
 00000000 0000 154 :  
 00000000 0000 155 YEAR=0 :  
 00000000 0000 156 MONTH=2 :  
 00000000 0000 157 DAY=4 :  
 00000000 0000 158 HOUR=6 :  
 00000000 0000 159 MINUTE=8 :  
 00000000 0000 160 SECOND=10 :  
 00000000 0000 161 HUNDREDTH=12 :  
 00000000 0000 162 :  
 00000000 0000 163 :  
 00000000 0000 164 : LOCAL DATA  
 00000000 0000 165 :  
 00000000 0000 166 : MONTH, DAY CONVERSION TABLE  
 00000000 0000 167 :  
 00000000 0000 168 PSECT Y\$EXEPAGED :  
 00000000 0000 169 DATETABLE: : DATE CONVERSION TABLE  
 1F 0000 170 .BYTE 31 : JANUARY  
 1D 0001 171 .BYTE 29 : FEBRUARY

1F	0002	172	.BYTE	31	:MARCH
1E	0003	173	.BYTE	30	:APRIL
1F	0004	174	.BYTE	31	:MAY
1E	0005	175	.BYTE	30	:JUNE
1F	0006	176	.BYTE	31	:JULY
1F	0007	177	.BYTE	31	:AUGUST
1E	0008	178	.BYTE	30	:SEPTEMBER
1F	0009	179	.BYTE	31	:OCTOBER
1E	000A	180	.BYTE	30	:NOVEMBER
1F	000B	181	.BYTE	31	:DECEMBER
	000C	182			
	000C	183			
	000C	184	: MONTH CONVERSION TABLE		
	000C	185	:		
	000C	186			
	000C	187	MONTHTAB:		

4E	41	4A	03	000C	188	.ASCII	<3>/JAN/	:
42	45	46	03	0010	189	.ASCII	<3>/FEB/	
52	41	4D	03	0014	190	.ASCII	<3>/MAR/	
52	50	41	03	0018	191	.ASCII	<3>/APR/	
59	41	4D	03	001C	192	.ASCII	<3>/MAY/	
4E	55	4A	03	0020	193	.ASCII	<3>/JUN/	
4C	55	4A	03	0024	194	.ASCII	<3>/JUL/	
47	55	41	03	0028	195	.ASCII	<3>/AUG/	
50	45	53	03	002C	196	.ASCII	<3>/SEP/	
54	43	4F	03	0030	197	.ASCII	<3>/OCT/	
56	4F	4E	03	0034	198	.ASCII	<3>/NOV/	
43	45	44	03	0038	199	.ASCII	<3>/DEC/	
				003C	200			
				003C	201			
				003C	202	: HOURS, MINUTES, SECONDS, HUNDREDTHS CONVERSION TABLE		
				003C	203	:		
				003C	204			
				003C	205	TIMETABLE:		:TIME CONVERSION TABLE

64	003C	206	.BYTE	100	:HUNDREDTHS
3C	003D	207	.BYTE	60	:SECONDS
3C	003E	208	.BYTE	60	:MINUTES AND HOURS
	003F	209			
	003F	210			
	003F	211	: CONVERSION CONTROL STRINGS		
	003F	212	:		
	003F	213			

5A	34	21	2D	43	41	21	2D	57	53	32	21	003F	214	DATE: .ASCII	/!2SW-!AC-!4ZW/	:
							20	57	004B							
32	21	3A	57	5A	32	21	3A	57	5A	32	21	004D	215	DELTA: .ASCII	/!4SW/	:
							57	5A	32	21	2E	0052	216	TIME: .ASCII	/!2ZW:!2ZW:!2ZW.!2ZW/	:
												005E				

0065 218 .SBTTL CONVERT BINARY TIME TO ASCII STRING  
 0065 219 +  
 0065 220 EXE\$ASCTIM - CONVERT BINARY TIME TO ASCII STRING  
 0065 221  
 0065 222 THIS SERVICE PROVIDES THE CAPABILITY TO CONVERT AN ABSOLUTE OR DELTA  
 0065 223 TIME FROM 64-BIT FORMAT TO AN ASCII STRING.  
 0065 224  
 0065 225 INPUTS:  
 0065 226  
 0065 227 ATIMLEN(AP) = ADDRESS OF WORD TO RECEIVE OUTPUT LENGTH.  
 0065 228 ATIMBUF(AP) = ADDRESS OF OUTPUT BUFFER DESCRIPTOR.  
 0065 229 ATIMADR(AP) = ADDRESS OF 64-BIT TIME VALUE. IF ZERO, THEN THE CURRENT  
 0065 230 SYSTEM TIME IS USED. POSITIVE VALUES ARE INTERPRETED AS  
 0065 231 ABSOLUTE TIMES AND NEGATIVE VALUES AS DELTA TIMES.  
 0065 232 ACVTFLG(AP) = CONVERSION INDICATOR.  
 0065 233 LOW BIT CLEAR INDICATES BOTH DATE AND TIME ARE TO BE CON-  
 0065 234 VERTED.  
 0065 235 LOW BIT SET INDICATES ONLY TIME IS TO BE CONVERTED.  
 0065 236

## 0065 237 OUTPUTS:

0065 238 R0 LOW BIT CLEAR INDICATES FAILURE TO CONVERT TIME TO ASCII.  
 0065 239  
 0065 240 R0 = SSS\_ACCVIO - 64-BIT TIME VALUE OR OUTPUT BUFFER DESCRIPTOR  
 0065 241 CANNOT BE READ BY CALLING ACCESS MODE, OR OUTPUT BUFFER  
 0065 242 CANNOT BE WRITTEN BY CALLING ACCESS MODE.  
 0065 243  
 0065 244 R0 = SSS\_IVTIME - SPECIFIED DELTA TIME IS GREATER THAN 9999  
 0065 245 DAYS.  
 0065 246  
 0065 247 R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.  
 0065 248  
 0065 249 R0 = SSS\_NORMAL - NORMAL COMPLETION.  
 0065 250  
 0065 251 :-

0065 252  
 0065 253 EXE\$ASCTIM:: :CONVERT TIME TO ASCII  
 7E 08 BC 007C 0065 254 .WORD :ENTRY MASK  
 56 5E D0 0067 0067 255 MOVQ :SAVE OUTPUT BUFFER DESCRIPTOR  
 56 5E D0 006B 006B 256 MOVL :SAVE ADDRESS OF OUTPUT BUFFER DESCRIPTOR  
 55 5E J0 006E 006E 257 CLRL :CLEAR SPACE FOR LENGTH FROM FA0  
 55 5E J0 0070 0070 258 MOVL :SAVE ADDRESS OF LENGTH  
 52 52 D4 0073 0073 259 CLRL :ASSUME ABSOLUTE TIME SPECIFIED  
 53 OC AC 0075 0075 260 MOVL :GET ADDRESS OF 64-BIT TIME VALUE  
 07 13 0079 0079 261 BEQL :IF EQL NONE SPECIFIED  
 50 63 7D 007B 007B 262 MOVL :GET 64-BIT TIME VALUE  
 02 18 007E 007E 263 BGEQ :IF GEQ ABSOLUTE TIME  
 52 52 D6 0080 0080 264 INCL :INDICATE DELTA TIME  
 5E 10 C2 0082 0082 265 10\$: SUBL :ALLOCATE NUMERIC TIME BUFFER  
 54 5E D0 0085 0085 266 MOVL :SAVE ADDRESS OF NUMERIC TIME BUFFER  
 6F 50 E9 0093 0093 267 SNUMTIM\_S(R4),(R3) :CONVERT TIME TO NUMERIC FORMAT  
 0096 268 BLBC :IF LBC CONVERSION FAILURE  
 0096 269  
 0096 270 : CONVERT TIME TO ASCII FORMAT  
 0096 271  
 0096 272  
 0096 273  
 3E 10 AC E8 0096 274 BLBS ACVTFLG(AP),40\$ :IF LBS ONLY TIME IS TO BE CONVERTED

12 52 E8 009A 275 BLBS R2,20\$ ;IF LBS DELTA TIME SPECIFIED

009D 276

009D 277

009D 278 ; CONVERT DATE

009D 279 ;

009D 280

52 52 02 A4 3C 009D 281 MOVZWL MONTH(R4),R2 ;GET NUMERIC MONTH VALUE

FF62 CF42 DE 00A1 282 MOVAL W^MONTHTAB-4[R2],R2 ;GET ADDRESS OF MONTH COUNTED STRING

FF94 CF DF 00A7 283 PUSHAL W^DATE ;BUILD DESCRIPTOR FOR CONTROL STRING

0E DD 00AB 284 PUSHL #DELTA-DATE

06 11 00AD 285 BRB 30\$ ;

00AF 286

00AF 287 ;

00AF 288 ; CONVERT DELTA TIME

00AF 289 ;

00AF 290

FF9A CF DF 00AF 291 20\$: PUSHAL W^DELTA ;BUILD CONTROL STRING DESCRIPTOR

05 DD 00B3 292 PUSHL #TIME-DELTA

51 5E DO 00B5 293 30\$: MOVL SP,R1 ;COPY ADDRESS OF CONTROL STRING DESCRIPTOR

36 50 E9 00C8 294 \$FAO\_S (R1),(R5),(R6),DAY(R4),R2,YEAR(R4) ;CONVERT DELTA TIME OR DATE

66 65 A2 00CF 295 BLBC R0,60\$ ;IF LBC CONVERT FAILURE

27 15 00D2 296 SUBW (R5),(R6) ;ANY SPACE LEFT IN TIME BUFFER?

04 A6 65 C0 00D4 297 BLEQ 50\$ ;IF LEQ NO

00D8 298 ADDL (R5),4(R6) ;UPDATE TIME BUFFER ADDRESS

00D8 299

00D8 300 ;

00D8 301 ; CONVERT TIME

00D8 302 ;

00D8 303

FF76 CF DF 00D8 304 40\$: PUSHAL W^TIME ;BUILD CONTROL STRING DESCRIPTOR

13 DD 00DC 305 PUSHL #EXE\$ASCTIM-TIME

51 5E DO 00DE 306 MOVL SP,R1 ;COPY ADDRESS OF CONTROL STRING DESCRIPTOR

51 04 AC DO 00E1 307 \$FAO\_S (R1),2(R5),(R6),HOUR(R4) ;MINUTE(R4),SECOND(R4),HUNDREDTH(R4) ;

04 13 00FF 308 50\$: MOVL ATIMLEN(AP\$),R1 ;LENGTH ADDRESS SPECIFIED?

61 65 85 A1 0101 310 BEQL 60\$ ;IF EQL NO

04 0105 311 ADDW3 (R5)+,(R5),(R1) ;COMPUTE AND RETURN OUTPUT LENGTH

311 60\$: RET ;

```

0106 313 .SBTL CONVERT ASCII STRING TO BINARY TIME
0106 314 :+ EXE$BINTIM - CONVERT ASCII STRING TO BINARY TIME
0106 315 : THIS SERVICE PROVIDES THE CAPABILITY TO CONVERT AN ASCII STRING TO A
0106 316 : 64-BIT ABSOLUTE OR DELTA TIME.
0106 317 :
0106 318 : INPUTS:
0106 319 :
0106 320 : BTIMBUF(AP) = ADDRESS OF ASCII STRING DESCRIPTOR.
0106 321 : BTIMADR(AP) = ADDRESS TO STORE 64-BIT TIME VALUE.
0106 322 :
0106 323 :
0106 324 :
0106 325 : OUTPUTS:
0106 326 :
0106 327 : R0 LOW BIT CLEAR INDICATES FAILURE TO CONVERT TIME TO ASCII.
0106 328 :
0106 329 : R0 = SSS_IVTIME - ASCII STRING HAS INVALID SYNTAX OR TIME
0106 330 : COMPONENT IS OUT OF RANGE.
0106 331 :
0106 332 : R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
0106 333 :
0106 334 : R0 = SSS_NORMAL - NORMAL COMPLETION.
0106 335 :-
0106 336 :
0106 337 EXESBINTIM::: ;CONVERT ASCII STRING TO BINARY TIME
0106 338 .WORD ^M<R2,R3,R4,R5,R6,R7,R8>;ENTRY MASK
0106 339 SUBL #<<<7*2>+3>/4>*4,SP;ALLOCATE NUMERIC TIME BUFFER
0106 340 MOVL SP,R7;SAVE ADDRESS OF NUMERIC TIME BUFFER
0106 341 MOVQ @BTIMBUF(AP),R5;GET ADDRESS AND LENGTH OF ASCII STRING
0106 342 CLRL R8;ASSUME DELTA TIME
0106 343 10$: DECW R5;ANY MORE CHARACTERS?
0106 344 BLSS 30$;IF LSS NO
0106 345 CMPB #BLANK,(R6)+;SKIP LEADING BLANK?
0106 346 BEQL 10$;IF EQL YES
0106 347 INCW R5;CORRECT NUMBER OF CHARACTERS
0106 348 LOCC #HYPHEN,R5,-(R6);ABSOLUTE TIME FORMAT?
0106 349 BEQL 30$;IF EQL NO
0106 350 INCL R8;INDICATE ABSOLUTE TIME
0106 351 $NUMTIM_S(R7);CONVERT CURRENT TIME TO NUMERIC FORMAT
0106 352 :
0106 353 :
0106 354 : CONVERT ABSOLUTE TIME
0106 355 :
0106 356 :
0106 357 MOVAL DAY(R7),R4;SET ADDRESS TO STORE DAY
0106 358 BSBW CONVERT;CONVERT DAY FIELD
0106 359 .BYTE HYPHEN;EXPECTED TERMINATOR
0106 360 TSTW R5;ANY MORE CHARACTERS?
0106 361 BNEQ 11$;BRNCH IF THERE ARE MORE CHARACTERS.
0106 362 BRW CVRTIME;IF NO MORE CHARACTERS, CONVERT TIME.
0106 363 11$: CMPB #HYPHEN,(R6)+;MONTH FIELD VOID?
0106 364 BEQL 20$;IF EQL YES
0106 365 MATCHC #3,-(R6),#4*12,W^MONTHTAB;SEARCH FOR MONTH SUBSTRING MATCH
0106 366 BEQL 14$;SKIP ERROR BRANCH IF MATCH FOUND
0106 367 BRW IVTIME;IF NEQ NO MATCH FOUND
0106 368 14$: SUBL3 R2,#4*12,R2;CALCULATE CHARACTERS TO START OF SUBSTRING
0106 369 BITL #3,R2;MULTIPLE OF 4?
55 57 10 01FC 0106
55 57 57 004 0106
55 57 04 BC 0106
55 58 04 0106
55 58 07 0106
55 64 19 0106
86 20 91 0106
86 F7 13 0106
76 55 55 0106
76 57 20 0106
76 57 13 0106
76 58 06 0106
54 04 A7 0106
54 0081 0106
55 85 0106
55 03 12 0106
86 00DB 0106
86 2D 91 0106
86 2F 13 0106
FEBF CF 30 76 03 0106
FEBF CF 03 13 0106
52 0092 0106
52 30 52 C3 0106
52 52 03 D3 0106

```

02 A7 03 13 0159 370 BEQL 16\$ :BRANCH IF MULTIPLE OF 4  
 52 0086 31 015B 371 BRW IVTIME :IF NOT MULTIPLE OF 4, THEN ERROR  
 52 04 A7 015E 372 16\$: DIVW3 #4,R2,MONTH(R7) :CONVERT TO MONTH AND STORE  
 56 03 C0 0163 373 ADDL #3,R6 :UPDATE ADDRESS OF ASCII STRING  
 55 03 A2 0166 374 SUBW #3,R5 :UPDATE COUNT OF REMAINING CHARACTERS  
 79 19 0169 375 BLSS IVTIME :IF LSS INVALID SYNTAX  
 0? 14 016B 376 BGTR 18\$ :IF GTR CHARACTERS REMAINING  
 00AC 31 016D 377 BRW CVRTIME :OTHERWISE END OF STRING  
 86 2D 91 0170 378 18\$: CMPB #HYPHEN,(R6)+ :FIELD TERMINATED PROPERLY?  
 6F 12 0173 379 BNEQ IVTIME :IF NEQ NO  
 55 B7 0175 380 20\$: DECW R5 :DECREMENT COUNT OF REMAINING CHARACTERS  
 54 67 DE 0177 381 MOVAL YEAR(R7),R4 :SET ADDRESS TO STORE YEAR  
 0A 11 017A 382 BRB 40\$ :  
 017C 383 :  
 017C 384 : CONVERT DELTA TIME  
 017C 385 :  
 017C 386 :  
 017C 387 :  
 54 67 DE 017C 388 30\$: MOVAL YEAR(R7),R4 :GET ADDRESS TO STORE YEAR  
 84 D4 017F 389 CLR L (R4)+ :CLEAR YEAR AND MONTH  
 64 7C 0181 390 CLR Q (R4) :CLEAR DAY, HOUR, MINUTE, AND SECOND  
 0C A7 B4 0183 391 CLR W HUNDREDTH(R7) :CLEAR HUNDREDTH  
 62 10 0186 392 40\$: BSSBB CONVERT :CONVERT RELATIVE DAY OR YEAR FIELD  
 20 0188 393 .BYTE BLANK :EXPECTED TERMINATOR  
 55 B7 0189 394 50\$: DECW R5 :ANY REMAINING CHARACTERS?  
 03 18 018B 395 BGEQ 53\$ :BRANCH IF CHARACTERS REMAINING  
 008C 31 018D 396 BRW CVRTIME :ELSE GO PROCESS WHAT WE'VE GOT  
 86 20 91 0190 397 53\$: CMPB #BLANK,(R6)+ :NEXT CHARACTER BLANK?  
 F4 13 0193 398 BEQL 50\$ :IF EQL YES  
 56 D7 0195 399 DECL R6 :BACK UP TO NONBLANK CHARACTER  
 55 D6 0197 400 INCL R5 :ADJUST REMAINING CHARACTER COUNT  
 0199 401 :  
 0199 402 : CONVERT TIME  
 0199 403 :  
 0199 404 :  
 0199 405 :  
 54 06 A7 DE 0199 406 MOVAL HOUR(R7),R4 :SET ADDRESS TO STORE HOUR  
 48 10 019D 407 BSSBB CONVERT :CONVERT HOUR FIELD  
 3A 019F 408 .BYTE COLON :EXPECTED TERMINATOR  
 48 10 01A0 409 BSSBB CONVERT :CONVERT MINUTE FIELD  
 3A 01A2 410 .BYTE COLON :EXPECTED TERMINATOR  
 45 10 01A3 411 BSSBB CONVERT :CONVERT SECOND FIELD  
 2E 01A5 412 .BYTE PERIOD :EXPECTED TERMINATOR  
 01A6 413 :  
 01A6 414 :  
 01A6 415 :  
 01A6 416 :  
 01A6 417 :  
 53 03 D0 01A6 418 MOVL #3, R3 :Convert Hundredth Field  
 64 B4 01A9 419 CLR W (R4) :This must be done differently because  
 55 B7 01AB 420 70\$: DECW R5 :this is a fractional value.  
 2L 19 01AD 421 BLSS 80\$ :Establish max useable digits,  
 51 86 9A 01AF 422 MOVZBL (R6)+, R1 :including the rounding digit.  
 20 51 91 01B2 423 CMPB R1, #BLANK :Clear accumulated value.  
 24 13 01B5 424 BEQL 80\$ :Any more characters?  
 51 30 C2 01B7 425 SUBL #ONE, R1 :Branch if no more characters.  
 28 19 01BA 426 BLSS IVTIME :Get the next character.  
 :A blank marks the end of the field.  
 :Branch if at end of the field.  
 :Subtract out character bias.  
 :Branch if invalid character.

51 09 D1 01BC 427 CMPL #NINE-ONE, R1 ;Result value within digit range?  
 23 19 01BF 428 BLSS JVTIME ;Branch if invalid character.  
 0B 53 F5 01C1 429 SOBGR R3 73\$ ;Branch if using this digit directly.  
 E5 19 01C4 430 BLSS 70\$ ;Branch if ignoring this digit.  
 51 05 D1 01C6 431 CMPL #5 R1 ;Else digit as the rounding digit.  
 E0 14 01C9 432 BGTR 70\$ ;Branch if rounding has no effect.  
 64 B6 01CB 433 INCW (R4) ;If rounding up, do it.  
 DC 11 01CD 434 BRB 70\$ ;Then loop, but for a regular digit.  
 64 0A A4 01CF 435 73\$: MULW #10, (R4) ;multiply partial result by 10.  
 10 1D 01D2 436 BVS JVTIME ;An overflow means an invalid time.  
 64 51 A0 01D4 437 ADDW R1, (R4) ;Accumulate fractional value.  
 0B 1D 01D7 438 BVS JVTIME ;Overflow means invalid time.  
 D0 11 01D9 439 BRB 70\$ ;Loop till end occurs.  
 53 D7 01DB 440 80\$: DECL R3 ;Insure that truncated digits are  
 3D 15 01DD 441 80\$: BLEQ CVRTIME ;included as zeros in the final  
 64 0A A4 01DF 442 MULW #10, (R4) ;fractional (hundredths) field value.  
 F7 11 01E2 443 BRB 80\$ ;NB: this will always overflow a word  
 01E4 444 ;if the fractional field has a  
 01E4 445 ;resolution greater than thousandths.  
 01E4 446 ;  
 01E4 447 ;  
 01E4 448 : INVALID SYNTAX OR TIME COMPONENT  
 01E4 449 :  
 50 0184 8F 3C 01E4 450 01E4 ;  
 04 01E9 451 IVTIME: MOVZWL #SSS\_IVTIME, R0 ;SET INVALID TIME  
 01EA 452 RET ;  
 01EA 453 ;  
 01EA 454 ;  
 01EA 455 : SUBROUTINE TO CONVERT NUMERIC FIELD TO BINARY  
 01EA 456 ;  
 01EA 457 ;  
 01EA 458 CONVERT: ;  
 50 D4 01EA 459 CLRL R0 ;CONVERT FIELD  
 84 B5 01EC 460 10\$: TSTW (R4)+ ;CLEAR ACCUMULATED VALUE  
 55 B7 01EE 461 11\$: DECW R5 ;POINT PAST NEXT FIELD  
 2A 19 01F0 462 BLSS CVRTIME ;ANY MORE CHARACTERS?  
 51 86 9A 01F2 463 MOVZBL (R6)+, R1 ;IF LSS NO  
 51 91 01F5 464 CMPB R1, @(\$P) ;GET NEXT CHARACTER  
 1E 13 01F9 465 BEQL 20\$ ;EXPECTED TERMINATOR?  
 20 51 91 01FB 466 CMPB R1, #BLANK ;IF EQL YES  
 EE 13 01FE 467 BEQL 11\$ ;BLANK CHARACTER?  
 51 30 C2 0200 468 SUBL #ONE, R1 ;IGNORE BLANKS  
 DF 19 0203 469 BLSS JVTIME ;SUBTRACT OUT CHARACTER BIAS  
 51 09 D1 0205 470 CMPL #NINE-ONE, R1 ;IF LSS INVALID CHARACTER  
 DA 19 0208 471 BLSS JVTIME ;RESULT VALUE WITHIN RANGE?  
 50 0A A4 020A 472 MULW #10, R0 ;IF LSS INVALID CHARACTER  
 D5 1D 020D 473 BVS JVTIME ;MULTIPLY PARTIAL RESULT BY 10  
 50 51 A0 020F 474 ADDW R1, R0 ;IF VS INVALID TIME  
 D0 1D 0212 475 BVS JVTIME ;ACCUMULATE VALUE  
 74 50 B0 0214 476 MOVW R0, -(R4) ;IF VS INVALID TIME VALUE  
 D3 11 0217 477 BRB 10\$ ;STORE VALUE  
 6E D6 0219 478 20\$: INCL (SP) ;INCREMENT PAST TERMINATOR  
 05 021B 479 RSB ;  
 021C 480 ;  
 021C 481 ;  
 021C 482 : CHECK CONVERTED DATE AND TIME VALUES  
 021C 483 :

04 A7 270F 8F B1 021C 484  
00 1F 0222 486 CVRTIME:  
06 A7 18 B1 0224 487 CMPW #9999, DAY(R7)  
08 A7 BA 18 0228 488 BLSSU IVTIME  
0A A7 B4 18 022E 489 CMPW #24, HOUR(R7)  
55 04 A7 3C B1 0230 490 BLEQU IVTIME  
03 58 E8 1B 0234 491 CMPW #60, MINUTE(R7)  
0097 31 023D 492 BLEQU IVTIME  
0240 493 MOVZWL DAY(R7), R5  
0240 494 BLBS R8 5\$  
0240 495 BRW 40\$  
0240 496 :  
0240 497 :  
0240 498 :  
0240 499 : CONVERT YEARS TO QUADRICENTURIES, CENTURIES, QUADYEARS, YEARS  
0240 500 :  
0240 501 :  
50 50 F9BF A2 13 0240 502 5\$: BEQL IVTIME  
51 50 50 00000190 8F 78 024E 503 MOVZWL YEAR(R7), R0  
52 51 51 00000064 8F 7B 0257 504 MOVAW -1601(R0), R0  
53 52 52 04 7B 0259 505 BLSS IVTIME  
0262 506 CLRL R1  
0264 507 EDIV #400, R0, R0, R1  
0269 508 CLRL R2  
0269 509 EDIV #100, R1, R1, R2  
0269 510 CLRL R3  
0269 511 EDIV #4, R2, R2, R3  
0269 512 :  
0269 513 :  
0269 514 : CONVERT QUADRICENTURIES, CENTURIES, QUADYEARS, YEARS TO DAYS  
0269 515 :  
0269 516 :  
52 53 52 53 016D 8F A4 0269 517 MULW #365, R3  
51 51 50 000005B5 8F 7A 026E 518 EMUL #QUADYEARDAYS, R2, R3, R2  
55 51 50 00023AB1 8F C4 0277 519 MULL #CENTURYDAYS, R1  
56 02 A7 3C 0289 520 EMUL #QUADRIDAYS, R0, R1, R5  
52 FD6B F40 50 52 028D 521 CLRL R0  
52 55 52 0290 522 MOVZWL MONTH(R7), R6  
50 50 1E 12 0296 523 10\$: ADDL R2, R5  
53 67 3C 0298 524 MOVZBL W^DATETABLE[R0], R2  
53 03 D3 029E 525 CMPL #1, R0  
14 12 02A1 526 BNEQ 30\$  
54 53 53 00000064 8F 78 02A3 527 MOVZWL YEAR(R7), R3  
54 54 D4 02A5 531 BITL #3, R3  
07 12 02B0 532 BNEQ 20\$  
53 03 D5 02B2 533 CLRL R4  
02 13 02B5 534 EDIV #100, R3, R3, R4  
52 D7 02B7 535 TSTL R4  
50 D0 50 56 F2 02B9 536 20\$: BNEQ 30\$  
51 0184 8F 3C 02BD 537 DECL R2  
55 00016FEC 8F C2 02C2 538 AOBLS R6, R0, 10\$  
51 04 A7 3C 02C2 539 MOVZWL #SSS, IVTIME, R0  
55 00016FEC 8F C2 02C6 540 MOVZWL DAY(R7), R1  
SUBL #TIMOFF2, R5  
:DAY WITHIN UPPER LIMIT?  
:IF LSSU NO  
:HOUR WITHIN LIMITS?  
:IF LEQU NO  
:MINUTE WITHIN LIMITS?  
:IF LEQU NO  
:SECOND WITHIN LIMITS?  
:IF LEQU NO  
:GET DAY VALUE  
:IF LBS ABSOLUTE TIME  
:  
: IF EQL INVALID TIME  
:GET YEAR VALUE  
:CALCULATE YEARS PAST 1601  
:IF LSS INVALID TIME  
:CLEAR HIGH PART OF DIVIDEND  
:CALCULATE QUADRICENTURIES  
:CLEAR HIGH PART OF DIVIDEND  
:CALCULATE CENTURIES  
:CLEAR HIGH PART OF DIVIDEND  
:CALCULATE QUADYEARS AND YEARS  
:  
: CALCULATE NUMBER OF DAYS PAST LEAP YEAR  
:CALCULATE NUMBER OF QUADYEAR DAYS AND SUM  
:CALCULATE NUMBER OF CENTURY DAYS  
:CALCULATE NUMBER OF QUADRIDAYS AND SUM  
:CLEAR INITIAL LOOP INDEX  
:GET SPECIFIED MONTH VALUE  
:ACCUMULATE TOTAL DAYS  
:GET NUMBER OF DAYS IN MONTH  
:SECOND MONTH OF YEAR?  
:IF NEQ NO  
:GET SPECIFIED YEAR VALUE  
:YEAR MULTIPLE OF 4?  
:IF NEQ NO  
:CLEAR HIGH PART OF DIVIDEND  
:CALCULATE CENTURY AND YEAR IN CENTURY  
:YEAR MULTIPLE OF 100?  
:IF NEQ NO  
:YEAR MULTIPLE OF 400?  
:IF EQL YES  
:REDUCE NUMBER OF DAYS IN MONTH  
:ANY MORE DAYS TO ACCUMULATE?  
:ASSUME INVALID DAY OF MONTH  
:GET SPECIFIED DAY  
:SUBTRACT OUT NUMBER OF DAYS TO 17-NOV-1858

```

55 51 C0 02CD 541      ADDL   R1 R5      :CALCULATE TOTAL NUMBER OF DAYS
57 19 02D0 542      BLSS   60$:      :IF LSS INVALID TIME
52 51 D1 02D2 543      CMPL   R1 R2      :DAY WITHIN LIMITS?
52 1A 02D5 544      BGTRU  60$:      :IF GTRU NO
52 02D7 545
52 02D7 546      : CONVERT TIME TO TENTHS OF MICROSECONDS
52 02D7 547
52 02D7 548
52 02D7 549
50 51 50 06 A7 3C 02D7 550 40$: MOVZWL HOUR(R7),R0      :GET HOUR VALUE
51 08 A7 3C 02D8 551      MOVZWL MINUTE(R7),R1      :GET MINUTE VALUE
50 51 50 3C 7A 02DF 552      EMUL   #60,R0,R1,R0      :CONVERT HOURS TO MINUTES AND SUM
51 0A A7 3C 02E4 553      MOVZWL SECOND(R7),R1      :GET SECOND VALUE
50 51 50 3C 7A 02E8 554      EMUL   #60,R0,R1,R0      :CONVERT MINUTES TO SECONDS AND SUM
51 0C A7 3C 02ED 555      MOVZWL HUNDREDTH(R7),R1      :GET HUNDREDTH VALUE
50 51 50 00000064 8F 7A 02F1 556      EMUL   #100,R0,R1,R0      :CONVERT SECONDS TO HUNDREDTHS AND SUM
50 00 50 000186A0 8F 7A 02FA 557      EMUL   #100000,R0,#0,R0      :CONVERT TO TENTHS OF MICROSECONDS
50 0303 558
50 0303 559      : CONVERT DAYS TO TENTHS OF MICROSECONDS
50 0303 560
50 0303 561
52 00 55 324A9A70 8F 7A 0303 562      EMUL   #843750000,R5,#0,R2      :MULTIPLY BY 864000000000/1024
52 52 0A 79 030C 563      ASHQ   #10,R2,R2      :MULTIPLY BY 1024
50 0310 564
50 0310 565
50 0310 566      : COMBINE RESULTS AND STORE 64-BIT TIME
50 0310 567
50 0310 568
50 0310 569
52 50 C0 0310 570      ADDL   R0,R2      :ADD LOW ORDER PARTS
53 51 D8 0313 571      ADWC   R1,R3      :ADD HIGH ORDER PARTS
50 01 3C 0316 572      MOVZWL #$$$ NORMAL,R0      :SET NORMAL COMPLETION
09 58 E8 0319 573      BLBS   R8,50$:      :IF LBS ABSOLUTE TIME
53 53 CE 031C 574      MNEGL  R3,R3      :CONVERT TO DELTA TIME
52 52 CE 031F 575      MNEGL  R2,R2
53 00 D9 0322 576      SBWC   #0,R3
08 BC 52 7D 0325 577 50$: MOVQ   R2,@BTIMADR(AP)      :STORE 64-BIT TIME VALUE
04 0329 578 60$: RET

```

032A 580 .SBTTL CONVERT BINARY TIME TO NUMERIC TIME  
 032A 581 .+ EXE\$NUMTIM - CONVERT BINARY TIME TO NUMERIC TIME  
 032A 582 : THIS SERVICE PROVIDES THE CAPABILITY TO CONVERT AN ABSOLUTE OR DELTA TIME  
 032A 583 : FROM 64-BIT FORMAT TO INTEGER DATE AND TIME VALUES.  
 032A 584 :  
 032A 585 :  
 032A 586 :  
 032A 587 :  
 032A 588 :  
 032A 589 :  
 032A 590 :  
 032A 591 :  
 032A 592 :  
 032A 593 :  
 032A 594 :  
 032A 595 :  
 032A 596 :  
 032A 597 :  
 032A 598 :  
 032A 599 :  
 032A 600 :  
 032A 601 :  
 032A 602 :  
 032A 603 :  
 032A 604 :  
 032A 605 :  
 032A 606 :  
 032A 607 :  
 032A 608 :  
 032A 609 :  
 032A 610 :  
 032A 611 :  
 032A 612 :  
 032A 613 :  
 032A 614 :  
 032A 615 :  
 032A 616 :  
 032A 617 :  
 032A 618 :  
 032A 619 :  
 032A 620 :  
 032A 621 :  
 032A 622 :  
 032A 623 :  
 032A 624 :  
 032A 625 :  
 032A 626 :  
 032A 627 :  
 032A 628 :  
 032A 629 :  
 032A 630 :  
 032A 631 :  
 032A 632 :  
 032A 633 :  
 032A 634 :  
 032A 635 :  
 032A 636 :  
 .SBTTL CONVERT BINARY TIME TO NUMERIC TIME  
 EXE\$NUMTIM - CONVERT BINARY TIME TO NUMERIC TIME  
 THIS SERVICE PROVIDES THE CAPABILITY TO CONVERT AN ABSOLUTE OR DELTA TIME  
 FROM 64-BIT FORMAT TO INTEGER DATE AND TIME VALUES.  
 INPUTS:  
 NTIMBUF(AP) = ADDRESS OF 7-WORD BUFFER TO RECEIVE CONVERTED DATE AND  
 TIME VALUES.  
 NTIMADR(AP) = ADDRESS OF 64-BIT TIME VALUE. IF ZERO, THEN THE CURRENT  
 SYSTEM TIME IS USED. POSITIVE VALUES ARE INTERPRETED AS  
 ABSOLUTE TIMES AND NEGATIVE VALUES AS DELTA TIMES.  
 OUTPUTS:  
 R0 LOW BIT CLEAR INDICATES FAILURE TO CONVERT TO NUMERIC TIME.  
 R0 = SSS\_ACCVIO - 64-BIT TIME VALUE CANNOT BE READ BY CALLING  
 ACCESS MODE OR TIME BUFFER CANNOT BE WRITTEN BY  
 CALLING ACCESS MODE.  
 R0 = SSS\_IVTIME - SPECIFIED DELTA TIME IS GREATER THAN 9999  
 DAYS.  
 R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.  
 R0 = SSS\_NORMAL - NORMAL COMPLETION.

57 04 AC 00FC 00:  
 50 01 3C 0336 0330 0339 0340 0347 0349 0350 0352 0356 0358 035E 0361 0363 0366 0369 036C 0370 0373 04:  
 51 00000000'EF 7D 0336 0339 0340 0347 0349 0350 0352 0356 0358 035E 0361 0363 0366 0369 036C 0370 0373 04:  
 51 00000000'EF D1 0336 0339 0340 0347 0349 0350 0352 0356 0358 035E 0361 0363 0366 0369 036C 0370 0373 04:  
 52 00000004'EF D1 0336 0339 0340 0347 0349 0350 0352 0356 0358 035E 0361 0363 0366 0369 036C 0370 0373 04:  
 52 F0 12 0347 0349 0350 0352 0356 0358 035E 0361 0363 0366 0369 036C 0370 0373 04:  
 53 08 AC 00: 0347 0349 0350 0352 0356 0358 035E 0361 0363 0366 0369 036C 0370 0373 04:  
 53 1C 13 0347 0349 0350 0352 0356 0358 035E 0361 0363 0366 0369 036C 0370 0373 04:  
 51 63 7D 035E 0361 0363 0366 0369 036C 0370 0373 04:  
 51 11 18 0361 0363 0366 0369 036C 0370 0373 04:  
 52 52 CE 0363 0366 0369 036C 0370 0373 04:  
 51 51 CE 0366 0369 036C 0370 0373 04:  
 52 00 D9 0369 036C 0370 0373 04:  
 04 50 00 E4 036C 0370 0373 04:  
 50 0C 3C 0370 0373 04:  
 04 0373 04:  
 .SBTTL CONVERT BINARY TIME TO NUMERIC TIME  
 EXE\$NUMTIM - CONVERT BINARY TIME TO NUMERIC TIME  
 THIS SERVICE PROVIDES THE CAPABILITY TO CONVERT AN ABSOLUTE OR DELTA TIME  
 FROM 64-BIT FORMAT TO INTEGER DATE AND TIME VALUES.  
 INPUTS:  
 NTIMBUF(AP) = ADDRESS OF 7-WORD BUFFER TO RECEIVE CONVERTED DATE AND  
 TIME VALUES.  
 NTIMADR(AP) = ADDRESS OF 64-BIT TIME VALUE. IF ZERO, THEN THE CURRENT  
 SYSTEM TIME IS USED. POSITIVE VALUES ARE INTERPRETED AS  
 ABSOLUTE TIMES AND NEGATIVE VALUES AS DELTA TIMES.  
 OUTPUTS:  
 R0 LOW BIT CLEAR INDICATES FAILURE TO CONVERT TO NUMERIC TIME.  
 R0 = SSS\_ACCVIO - 64-BIT TIME VALUE CANNOT BE READ BY CALLING  
 ACCESS MODE OR TIME BUFFER CANNOT BE WRITTEN BY  
 CALLING ACCESS MODE.  
 R0 = SSS\_IVTIME - SPECIFIED DELTA TIME IS GREATER THAN 9999  
 DAYS.  
 R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.  
 R0 = SSS\_NORMAL - NORMAL COMPLETION.

032A 611 EXE\$NUMTIM:  
 032A 612 .WORD ^M<R2,R3,R4,R5,R6,R7>  
 032A 613 MOVL NTIMBUF(AP),R7  
 032A 614 IFNOWRT #7\*2,(R7),10\$  
 032A 615 MOVZWL #SSS\_NORMAL,RO  
 032A 616 5\$: MOVQ EXESGQ\_SYSTIME,R1  
 032A 617 CMPL EXESGQ\_SYSTIME,R1  
 032A 618 BNEQ 5\$  
 032A 619 CMPL EXESGQ\_SYSTIME+4,R2  
 032A 620 BNEQ 5\$  
 032A 621 MOVL NTIMADR(AP),R3  
 032A 622 BEQL 20\$  
 032A 623 IFNORD #8,(R3),10\$  
 032A 624 MOVQ (R3),R1  
 032A 625 BGEQ 20\$  
 032A 626 MNEG L R2,R2  
 032A 627 MNEG L R1,R1  
 032A 628 SBWC #0,R2  
 032A 629 BBSC #0,RO,20\$  
 032A 630 10\$: MOVZUL #SSS\_ACCVIO,RO  
 032A 631 RET  
 032A 632 :  
 032A 633 :  
 032A 634 :  
 032A 635 :  
 032A 636 :  
 .SBTTL CONVERT BINARY TIME TO NUMERIC TIME  
 EXE\$NUMTIM - CONVERT BINARY TIME TO NUMERIC TIME  
 THIS SERVICE PROVIDES THE CAPABILITY TO CONVERT AN ABSOLUTE OR DELTA TIME  
 FROM 64-BIT FORMAT TO INTEGER DATE AND TIME VALUES.  
 INPUTS:  
 NTIMBUF(AP) = ADDRESS OF 7-WORD BUFFER TO RECEIVE CONVERTED DATE AND  
 TIME VALUES.  
 NTIMADR(AP) = ADDRESS OF 64-BIT TIME VALUE. IF ZERO, THEN THE CURRENT  
 SYSTEM TIME IS USED. POSITIVE VALUES ARE INTERPRETED AS  
 ABSOLUTE TIMES AND NEGATIVE VALUES AS DELTA TIMES.  
 OUTPUTS:  
 R0 LOW BIT CLEAR INDICATES FAILURE TO CONVERT TO NUMERIC TIME.  
 R0 = SSS\_ACCVIO - 64-BIT TIME VALUE CANNOT BE READ BY CALLING  
 ACCESS MODE OR TIME BUFFER CANNOT BE WRITTEN BY  
 CALLING ACCESS MODE.  
 R0 = SSS\_IVTIME - SPECIFIED DELTA TIME IS GREATER THAN 9999  
 DAYS.  
 R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.  
 R0 = SSS\_NORMAL - NORMAL COMPLETION.

					0374	637 : THE DIVISION IS PERFORMED IN THREE STEPS TO INSURE BOTH QUOTIENT AND
					0374	638 : REMAINDER STAY WITHIN 32 BITS.
					0374	639 :
					0374	640 : CALCULATE DAYS BY DIVIDING BY 1024 AND THEN 843750000. QUOTIENT IS DAYS
					0374	641 : AND REMAINDER IS FRACTION OF DAY.
					0374	642 :
					0374	643 :
52	51	54 51 0A 00	EF	79	0374	644 20\$: EXTZV #0 #10,R1,R4 :SAVE REMAINDER FROM NEXT DIVIDE
		51 51 51 FC 8F		7B	0379	645 ASHQ #-10 R1 R1 :DIVIDE BY 1024
		324A9A70 8F			037E	646 EDIV #843750000,R1,R1,R2 :CALCULATE DAYS AND FRACTION OF DAY
					0387	647 :
					0387	648 :
					0387	649 : R1 CONTAINS DAYS PAST BASE TIME. R2 PLUS R4 CONTAIN FRACTION OF DAY.
					0387	650 : R2 CONTAINS PART OF FRACTION IN UNITS OF 864000000000/1024 AND
					0387	651 : R4 CONTAINS REMAINDER IN UNITS OF TENTHS OF MICROSECONDS.
					0387	652 :
					0387	653 : CALCULATE FRACTION OF DAY IN HUNDRETHS OF SECONDS BY DIVIDING BY
					0387	654 : 100000 WHICH IS THE NUMBER OF TENTHS OF MICROSECONDS IN A HUNDRETH
					0387	655 : OF A SECOND.
					0387	656 :
					0387	657 :
52	52	52 52 0A 53	D4	79	0387	658 CLRL R3 :CLEAR HIGH PART OF DIVIDEND
	55	52 52 54 00	C8	0389	659 ASHQ #10,R2,R2 :CONVERT BACK TO TENTHS OF MICROSECONDS	
		000186A0 8F	7B	038D	660 BISL R4 R2 :ADD REMAINDER BACK	
				0390	661 EDIV #100000,R2,R5,R2 :CALCULATE FRACTION OF DAY IN HUNDRETHS	
				0399	662 :	
				0399	663 :	
				0399	664 : R1 CONTAINS DAYS PAST THE BASE TIME AND R5 CONTAINS THE FRACTION OF DAY	
				0399	665 : IN HUNDRETHS OF A SECOND.	
				0399	666 :	
		7E 50 00	E3	0399	667 BBCS #0,R0,70\$ :IF CLR, DELTA TIME SPECIFIED	
				0390	668 :	
				0390	669 :	
				0390	670 : ADD TIME OFFSET SO THAT DAY IS RELATIVE TO 1-JAN-1501.	
				0390	671 :	
51	0001FE98 8F	C0	0390	672 :		
				03A4	673 ADDL #TIMOFF1,R1 :ADD TIME OFFSET	
				03A4	674 :	
				03A4	675 :	
				03A4	676 : CALCULATE NUMBER OF QUADRICEENTURIES THAT HAVE PAST SINCE 1501.	
				03A4	677 :	
				03A4	678 :	
				03A4	679 :	
52	51	51 00023AB1 52	D4	03A4	680 CLRL R2 :CLEAR HIGH PART OF DIVIDEND	
		00023AB1 8F	7B	03A6	681 EDIV #QUADRIDIAYS,R1,R1,R2 :CALCULATE NUMBER OF QUADRICEENTURIES	
				03AF	682 :	
				03AF	683 :	
				03AF	684 : R1 CONTAINS THE NUMBER OF QUADRICEENTURIES AND R2 CONTAINS THE NUMBER OF	
				03AF	685 : DAYS INTO THE NEXT QUADRICEENTURY. CALCULATE THE NUMBER OF CENTURIES BY	
				03AF	686 : CONVERTING TO QUARTER DAYS INTO NEXT QUADRICEENTURY AND THEN DIVIDING BY	
				03AF	687 : THE AVERAGE NUMBER OF QUARTER DAYS IN A CENTURY.	
				03AF	688 :	
				03AF	689 :	
53	52	52 04 53 02	C4	03AF	690 MULL #4,R2 :CALCULATE NUMBER OF QUARTER DAYS	
		00023AB1 8F	03B2	691 CLRL R3 :CLEAR HIGH PART OF DIVIDEND		
			7B	0384	692 EDIV #QDAYSPCENT,R2,R2,R3 :CALCULATE NUMBER OF CENTURIES	
				03BD	693 :	

03BD 694  
 03BD 695  
 03BD 696  
 03BD 697  
 03BD 698  
 03BD 699  
 03BD 700  
 03BD 701  
 03BD 702  
 03BD 703  
 03BF 704  
 03C2 705  
 03CB 706  
 03CE 707  
 03D0 708  
 03D0 709  
 03D0 710  
 03D0 711  
 03D0 712  
 03D0 713  
 03D0 714  
 03D0 715  
 03D0 716  
 03D0 717  
 03D0 718  
 03D4 719  
 03D7 720  
 03DD 721  
 03E0 722  
 03E0 723  
 03E0 724  
 03E0 725  
 03E0 726  
 03E0 727  
 03E3 728  
 03E5 729  
 03F0 730  
 03F2 731  
 03F4 732  
 03F7 733  
 03F9 734  
 03FC 735 30\$:  
 03FE 736  
 03FE 737  
 0400 738 40\$:  
 0403 739 50\$:  
 0409 740  
 040C 741  
 040E 742  
 0412 743 60\$:  
 0415 744  
 0419 745  
 041B 746  
 041B 747  
 041B 748  
 041B 749  
 041B 750

R2 CONTAINS THE NUMBER OF CENTURIES AND R3 CONTAINS THE NUMBER OF QUARTER DAYS INTO THE NEXT CENTURY.  
 CALCULATE YEARS BY DISCARDING ANY FRACTION OF A DAY, ADDING 3/4'THS OF A DAY, AND DIVIDING BY THE AVERAGE NUMBER OF DAYS IN A YEAR. THE LEAP DAY OF EACH FOUR YEAR CYCLE IS FORCED INTO THE FOURTH YEAR.  
 CLRL R4  
 BISL #3,R3  
 EDIV #0DAYSPYEAR,R3,R3,R4  
 DIVL #4,R4  
 INCL R4

;CLEAR HIGH PART OF DIVIDEND  
 ;TRUNCATE FRACTION AND ADD 3/4'THS OF DAY  
 ;CALCULATE NUMBER OF YEARS  
 ;CALCULATE NUMBER OF DAYS MINUS ONE  
 ;CONVERT TO ACTUAL JULIAN DAY OF YEAR

R1 CONTAINS NUMBER OF QUADRICEENTURIES.  
 R2 CONTAINS NUMBER OF CENTURIES.  
 R3 CONTAINS NUMBER OF YEARS.  
 R4 CONTAINS JULIAN DAY OF YEAR.  
 CALCULATE ACTUAL YEAR.

MOVAL (R2)[R1],R1  
 MULL #50,R1  
 MOVAW 1501(P3)[R1],R1  
 MOVW R1,(R7)+  
 TEST FOR NONLEAP YEAR AND BIAS DAY IF AFTER 28-FEB.

BITL #3,R1  
 BNEQ 30\$  
 CLRL R2  
 EDIV #100,R1,R1,R2  
 TSTL R2  
 BNEQ 40\$  
 BITL #3,R1  
 BEQL 40\$  
 CMPL #31+28,R4  
 BGEO 40\$  
 INCL R4  
 MOVL #1,R1  
 MOVZBL W^DATETABLE-1[R1],R2  
 SUBL R2,R4  
 BLEQ 60\$  
 AOBLEQ #12,R1,50\$  
 MOVW R1,(R7)+  
 ADDW3 R2,R4,(R7)+  
 BRB 80\$  
 ;YEAR MULTIPLE OF 4?  
 ;IF NEQ NO  
 ;CLEAR HIGH PART OF DIVIDEND  
 ;CALCULATE CENTURY AND YEAR IN CENTURY  
 ;YEAR MULTIPLE OF 100?  
 ;IF NEQ NO  
 ;YEAR MULTIPLE OF 400?  
 ;IF EQL YES  
 ;AFTER 28-FEB?  
 ;IF GEQ NO  
 ;ADJUST FOR TABLE BIAS  
 ;INITIALIZE MONTH  
 ;GET NUMBER OF DAYS IN MONTH  
 ;SUBTRACT FROM JULIAN DAY  
 ;IF LEQ CORRECT MONTH FOUND  
 ;LOOP THROUGH ALL MONTHS  
 ;STORE MONTH  
 ;STORE DAY

DELTA TIME SPECIFIED - STORE RELATIVE DAY

87 51 D4 041B 751 70\$: CLRL (R7)+ :CLEAR YEAR AND MONTH  
 51 00002710 8F 51 B0 041D 752 MOVW R1 (R7)+ :STORE DAY  
 06 06 D1 0420 753 CMPL #10000,R1 :RELATIVE DAY WITHIN LIMITS?  
 50 0184 8F 1A 0427 754 BGTRU 80\$ :IF GTRU YES  
 04 3C 0429 755 MOVZWL #SSS\_IVTIME, R0 :SET INVALID TIME  
 04 042E 756 RET :  
 04 042F 757 :  
 04 042F 758 : R5 CONTAINS FRACTION OF DAY IN HUNDREDS OF SECONDS.  
 04 042F 760 :  
 04 042F 761 : CALCULATE HOUR, MINUTE, SECOND, AND HUNDRETH OF SECOND.  
 04 042F 762 :  
 04 042F 763 :  
 57 08 C0 042F 764 80\$: ADDL #8,R7 :POINT TWO BYTES PAST END OF BUFFER  
 51 51 D4 0432 765 CLRL R1 :CLEAR LOOP INDEX  
 52 FC03 CF41 9A 0434 766 90\$: MOVZBL W^TIMETABLE[R1],R2 :GET NEXT UNIT DIVISOR  
 56 55 55 56 D4 043A 767 CLRL R6 :CLEAR HIGH PART OF DIVIDEND  
 56 55 52 7B 043C 768 EDIV R2,R5,R5,R6 :CALCULATE NEXT PART  
 EC 51 02 F3 0444 769 MOVW R6,-(R7) :STORE NEXT PART  
 77 55 B0 0441 770 AOBLEQ #2,R1,90\$ :LOOP FOR HUNDREDS, SECONDS, AND MINUTES  
 04 0448 771 MOVW R5,-(R7) :STORE HOUR  
 04 0448 772 RET :  
 04 044C 773 :  
 04 044C 774 .END

SST2 = 00000007  
 ACVFLG = 00000010  
 ATIMADR = 0000000C  
 ATIMBUF = 00000008  
 ATIMLEN = 00000004  
 BLANK = 00000020  
 BTIMADR = 00000008  
 BTIMBUF = 00000004  
 CENTURYDAYS = 00008EAC  
 COLON = 0000003A  
 CONVERT 000001EA R 02  
 CVRTIME 0000021C R 02  
 DATE 0000003F R 02  
 DATETABLE = 00000000 R 02  
 DAY = 00000004  
 DELTA 00000040 R 02  
 EXE\$ASCTIM 00000065 RG 02  
 EXE\$BINTIM 00000106 RG 02  
 EXE\$GQ SYSTIME \*\*\*\*\* X 02  
 EXE\$NUMT 0000032A RG 02  
 HOUR = 00000006  
 HUNDREDTH = 0000000C  
 HYPHEN = 0000002D  
 IVTIME 000001E4 R 02  
 MINUTE = 00000008  
 MONTH = 00000002  
 MONTHTAB 0000000C R 02  
 NINE = 00000039  
 NTIMADR = 00000008  
 NTIMBUF = 00000004  
 ONE = 00000030  
 PERIOD = 0000002E  
 QDAYSPCENT = 00023AB1  
 QDAYSPYEAR = 000005B5  
 QUADRIDAYS = 00023AB1  
 QUADYEARDAYS = 000005B5  
 SECOND = 0000000A  
 SSS\$ACCVIO = 0000000C  
 SSS\$IVTIME = 00000184  
 SSS\$NORMAL = 00000001  
 SYS\$FAO \*\*\*\*\* X 02  
 SYS\$NUMT 00000001 \*\*\*\*\* GX 02  
 TIME 00000052 R 02  
 TIMETABLE 0000003C R 02  
 TIMOFF1 = 0001FE98  
 TIMOFF2 = 0001FEC  
 YEAR = 00000000

-----  
! Psect synopsis !  
-----

## PSECT name

## Allocation

## PSECT No.

## Attributes

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
YSEXEPAGED	0000044C ( 1100.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:01.01
Command processing	110	00:00:00.58	00:00:04.22
Pass 1	232	00:00:05.96	00:00:20.69
Symbol table sort	0	00:00:00.68	00:00:02.58
Pass 2	143	00:00:01.89	00:00:05.95
Symbol table output	8	00:00:00.07	00:00:00.32
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	525	00:00:09.28	00:00:34.79

The working set limit was 1500 pages.

34724 bytes (68 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 451 non-local and 39 local symbols.

774 source lines were read in Pass 1, producing 15 object records in Pass 2.

14 pages of virtual memory were used to define 12 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	2
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	9

505 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SYSCVRTIM/OBJ=OBJ\$:SYSCVRTIM MSRC\$:SYSCVRTIM/UPDATE=(ENH\$:SYSCVRTIM)+EXECMLS/LIB

0383 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY